# Accessibility Overview

## Session 009

# Accessibility Overview

**Travis W. Brown**
**Technology Evangelism**

Apple is committed
to making its products
accessible to all users

# Agenda

- Why is Accessibility important?
- Section 508
- Accessibility in Jaguar
    - Universal Access features
    - Developer Accessibility features
        - Assistive Technology
        - Jaguar Accessibility APIs

# Why Is Accessibility Important?

- 54 million* people in the US have disabilities
  - These are our mutual customers!
- Accessibility is a sales requirement
  - Federal, State, and Local
  - Education
  - Section 504 and 508
- Right thing to do!

*U.S. Dept. of Education

# Section 508

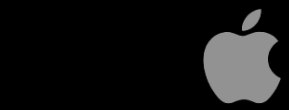
- In 1998, the federal government re-authorized Section 508 of the Rehabilitation Act
    - Sets down minimum accessibility requirements for operating systems, applications, websites and electronic devices
    - In effect as of June 21, 2001
- Section 508 is “the” Accessibility standard

# Interpreting Section 508

- Visit the Section 508 website:
  - **www.access-board.gov/508.htm**
  - Tip: Read the commentary documents
- Do not Panic!
  - Operating Systems and Assistive Technology play an important role

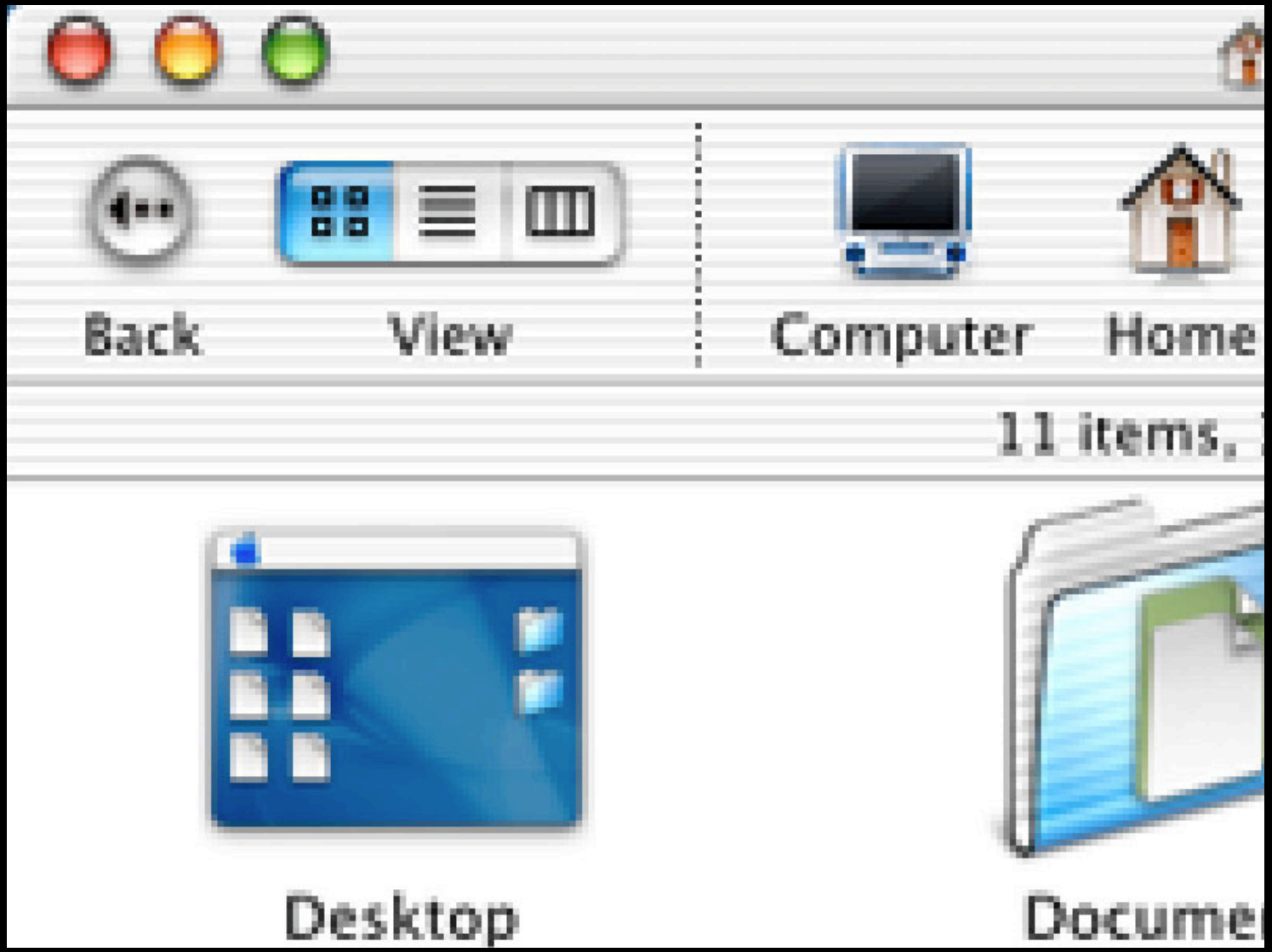
# Jaguar Accessibility

- User-Level Accessibility
  - Universal Access features
- Developer-Level Accessibility
  - Assistive Technology
  - Accessibility APIs



# Universal Access

# Zoom

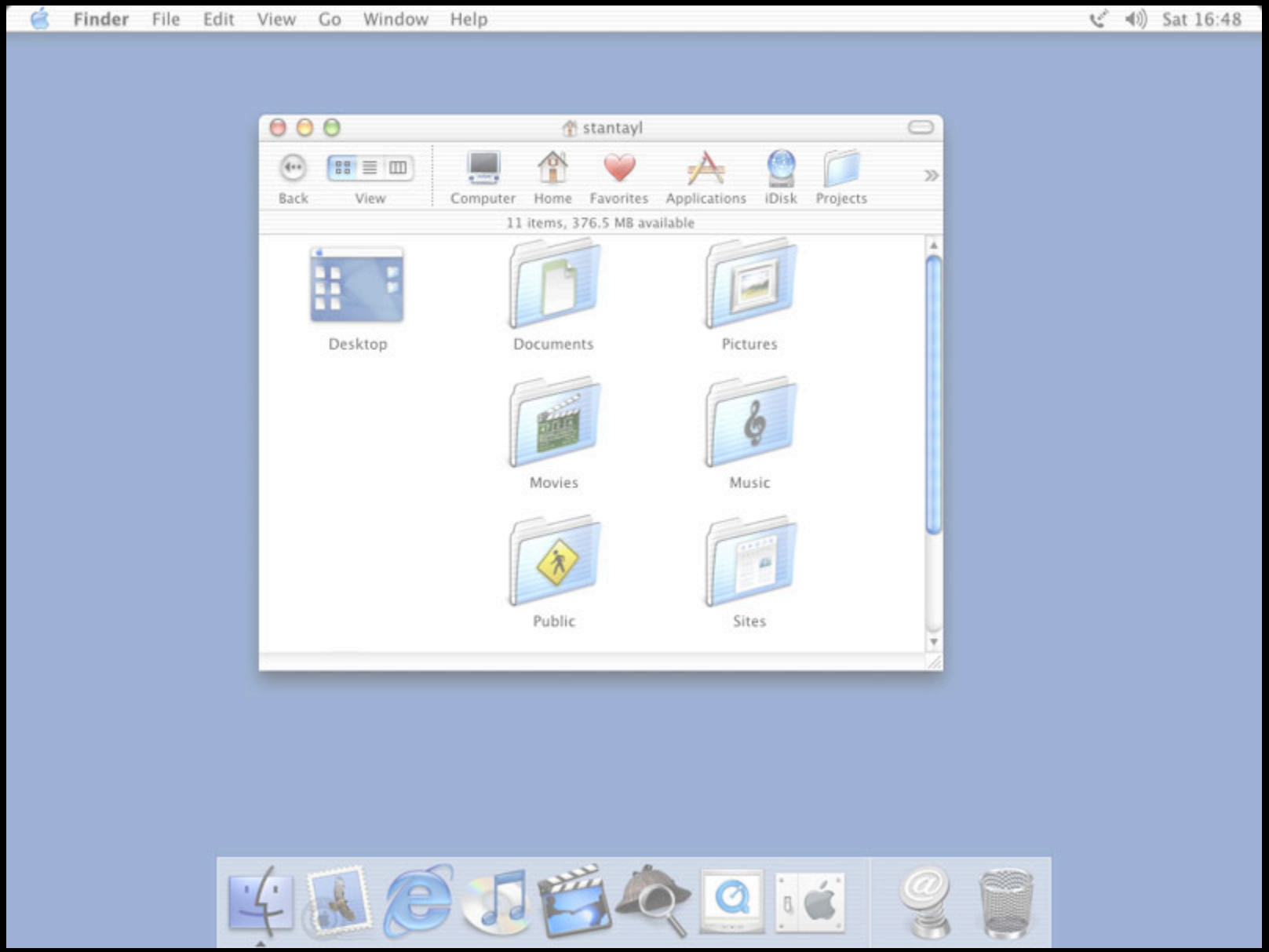


# Inversion


Finder File Edit View Go Window Help
Sat 16:48
stantayl
Back
View
Computer
Home
Favorites
Applications
iDisk
Projects
11 items, 376.5 MB available
Desktop
Documents
Pictures
Movies
Music
Public
Sites

# Visible Beep

# Visible Beep



# Visible Beep

# Sticky Keys


Finder File Edit View Go Window Help
Sat 16:48
stantayl
Back View
Computer Home Favorites Applications iDisk Projects
11 items, 376.5 MB available
Desktop
Documents
Pictures
Movies
Music
Public
Sites

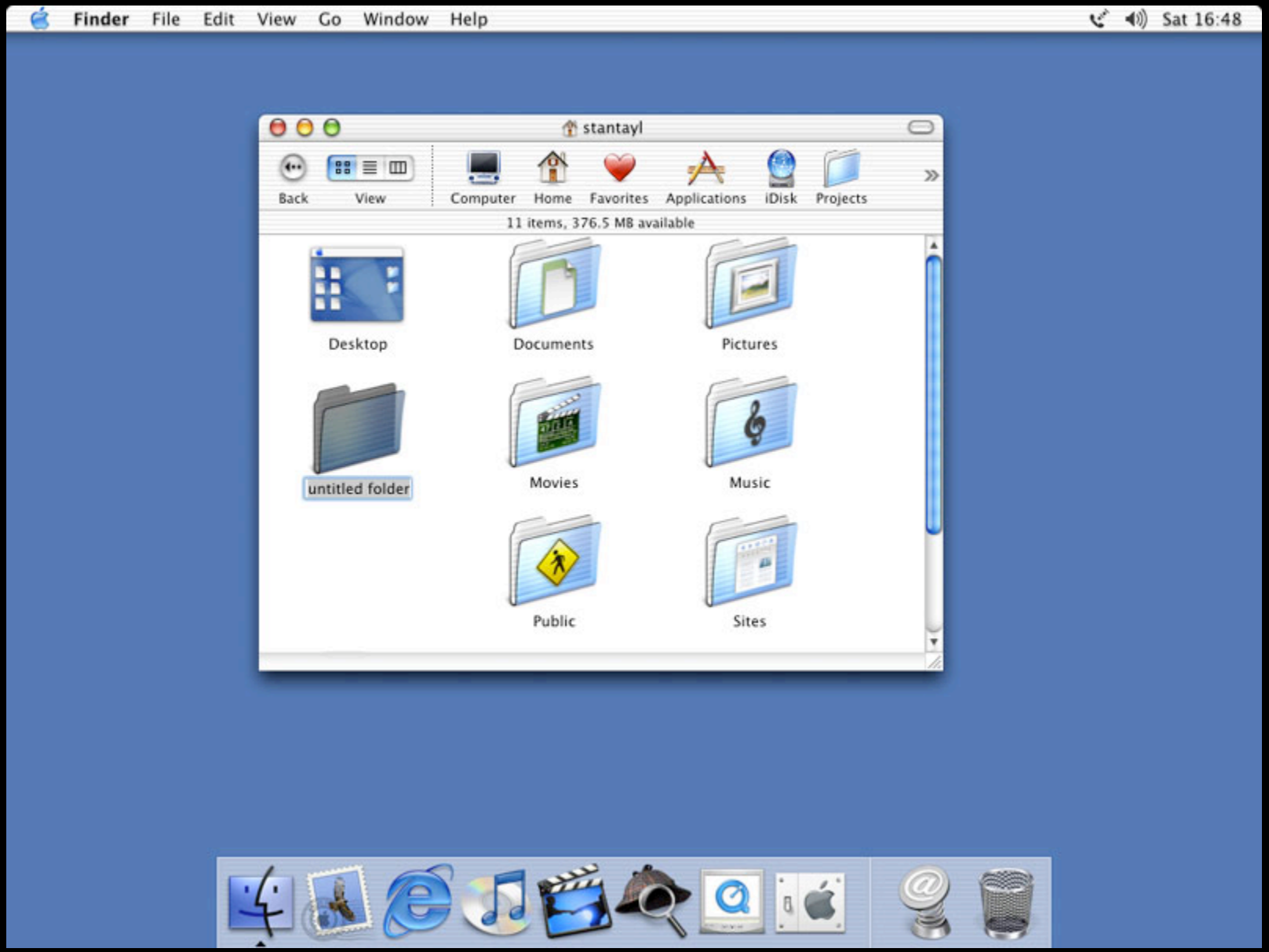
# Sticky Keys


Finder File Edit View Go Window Help
Sat 16:44
stantayl
Back View
Computer Home Favorites Applications iDisk Projects
10 items, 376.5 MB available
Desktop
Documents
Pictures
Movies
Music
Public
Sites

# Sticky Keys


Finder File Edit View Go Window Help
Sat 16:47
stantayl
Back View
Computer Home Favorites Applications iDisk Projects
10 items, 376.5 MB available
Desktop
Documents
Pictures
Movies
Music
Public
Sites

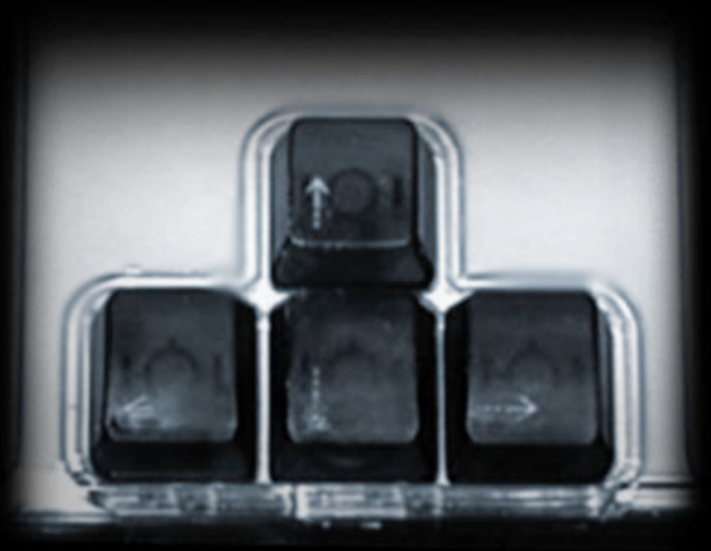
# Sticky Keys


Finder File Edit View Go Window Help
Sat 16:48
stantayl
Back View
Computer Home Favorites Applications iDisk Projects
11 items, 376.5 MB available
Desktop
Documents
Pictures
untitled folder
Movies
Music
Public
Sites

# Slow Keys

# Mouse Keys

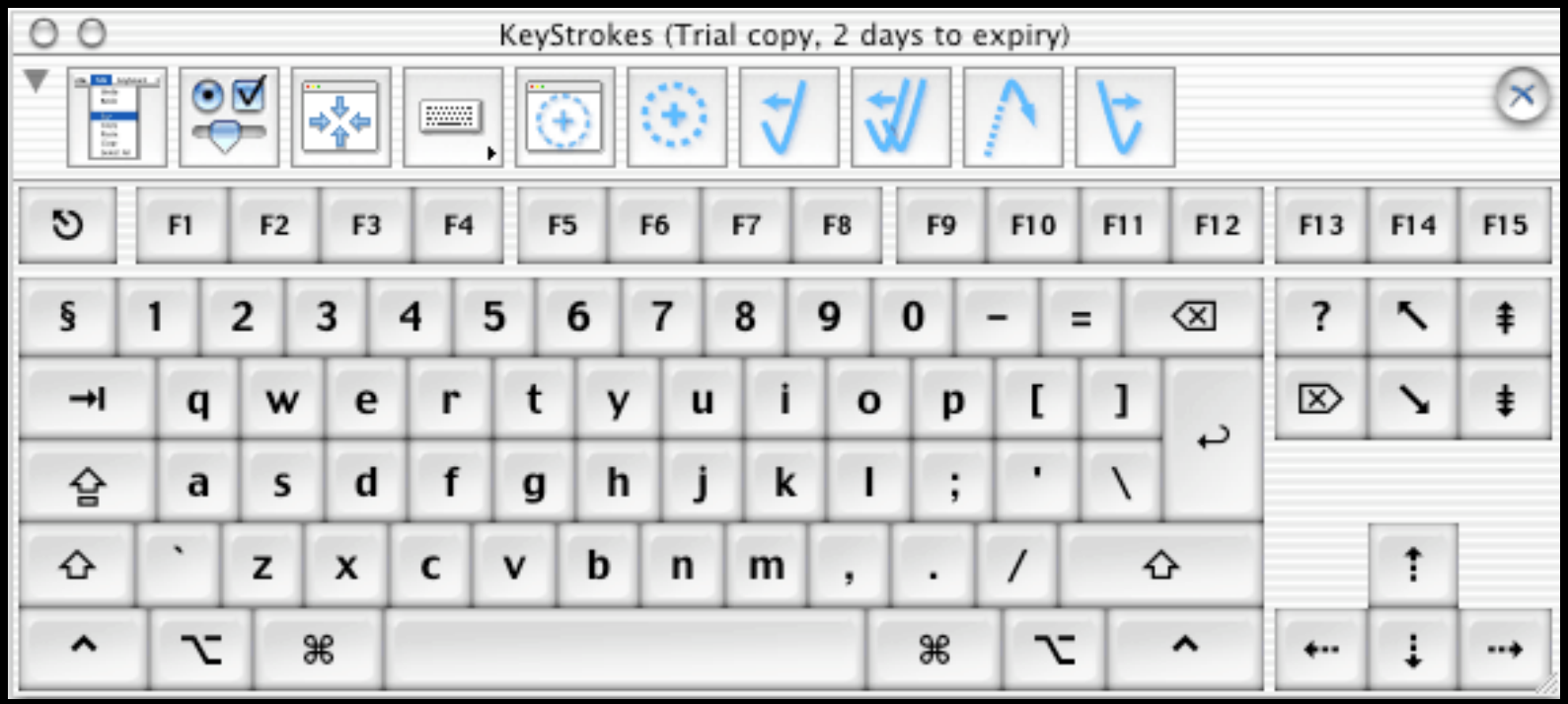
# Keyboard Navigation



File

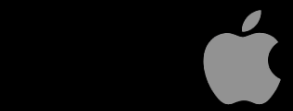
| | |
|---|---|
| New | ⌘N |
| Open... | ⌘O |
| Close | ⌘W |
| Save | ⌘S |
| Save As... | ⇧⌘S |
| Revert | ⌘R |
| Page Setup... | ⇧⌘P |
| Print... | ⌘P |

# Assistive Technology



# Assistive Technology Defined

- Assistive technology (AT) is any item, piece of equipment, or product system, whether acquired commercially off the shelf, modified,or customized, that is used to increase, maintain, or improve the functional capabilities of individuals with disabilities (29 U.S.C. Sec 2202(2))



# Assistive Tech and Section 508

- Section 508 also mandates low-level support for assistive technology
    - Keyboard navigation must be available
    - Expose the following UI information:
        - Control identity, state and operation
        - UI Focus state and location
        - On-screen text
        - Others . . .

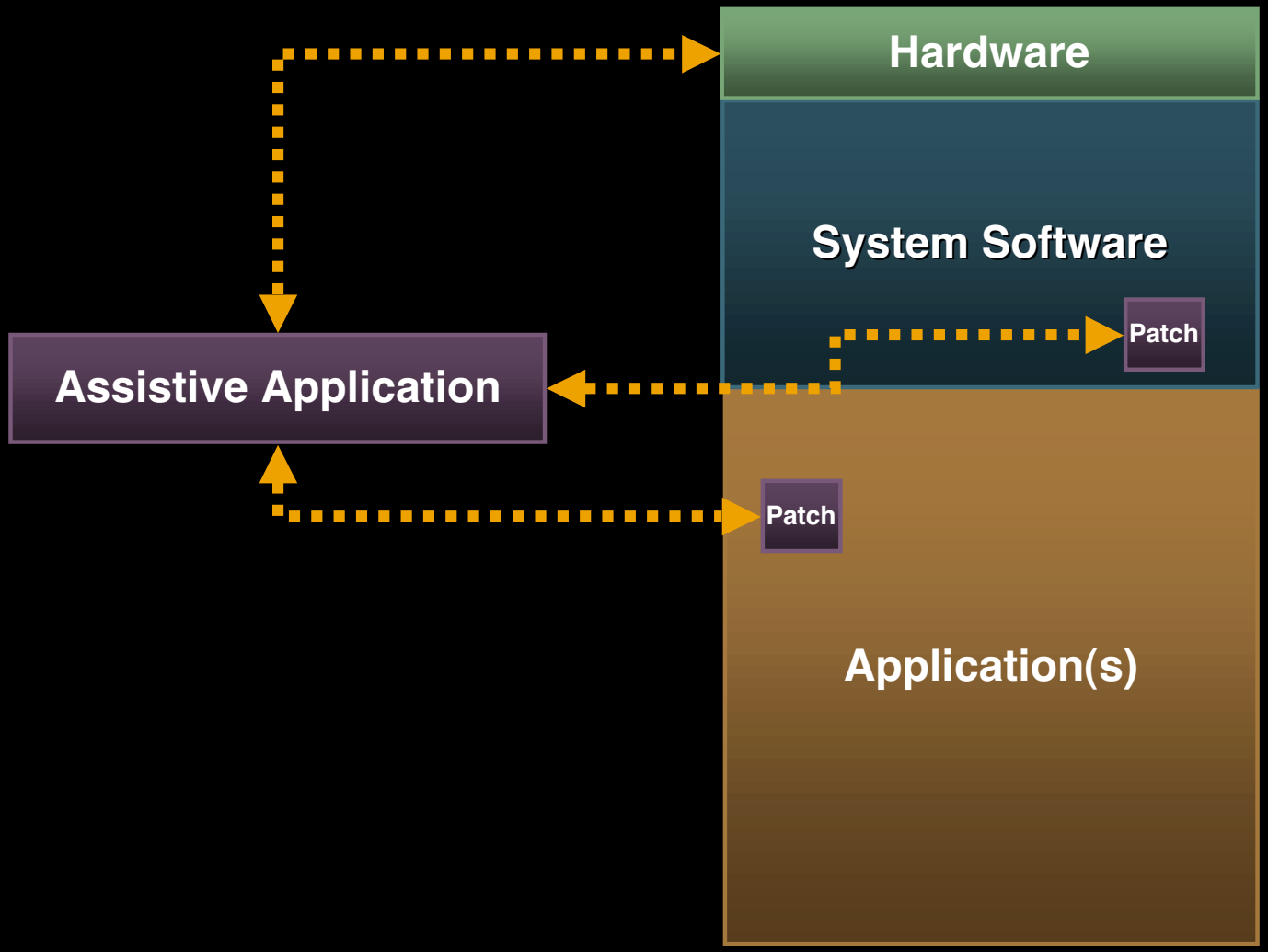
# Typical Assistive Technology

- Alternate input
  - Allow those unable to use standard input devices to interact with the system
  - Switches, eye tracking systems, alternate keyboards, command and control utilities, etc.
  - Usually a combination of hardware and software
    - Hardware interfaces
    - Synthesize events
    - UI discovery and control

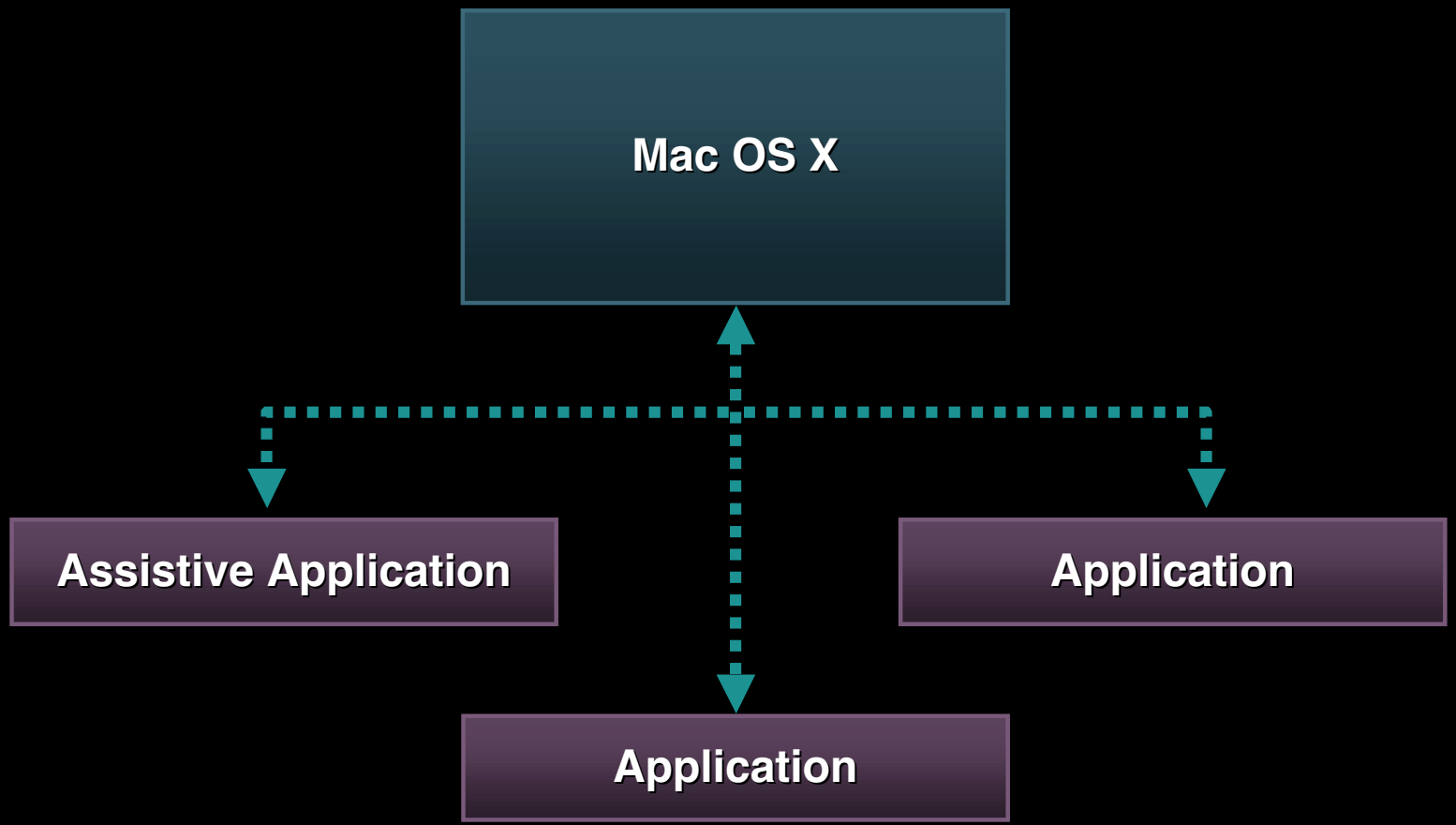
# Typical Assistive Technology

- Screen readers
  - Communicate the on-screen content, including text and UI elements, to user with vision disabilities
  - Text-to-speech and/or output to braille displays
  - Discover UI state, focus, and mouse location
  - Requires cooperation of assistive technology provider, application developers, and Apple

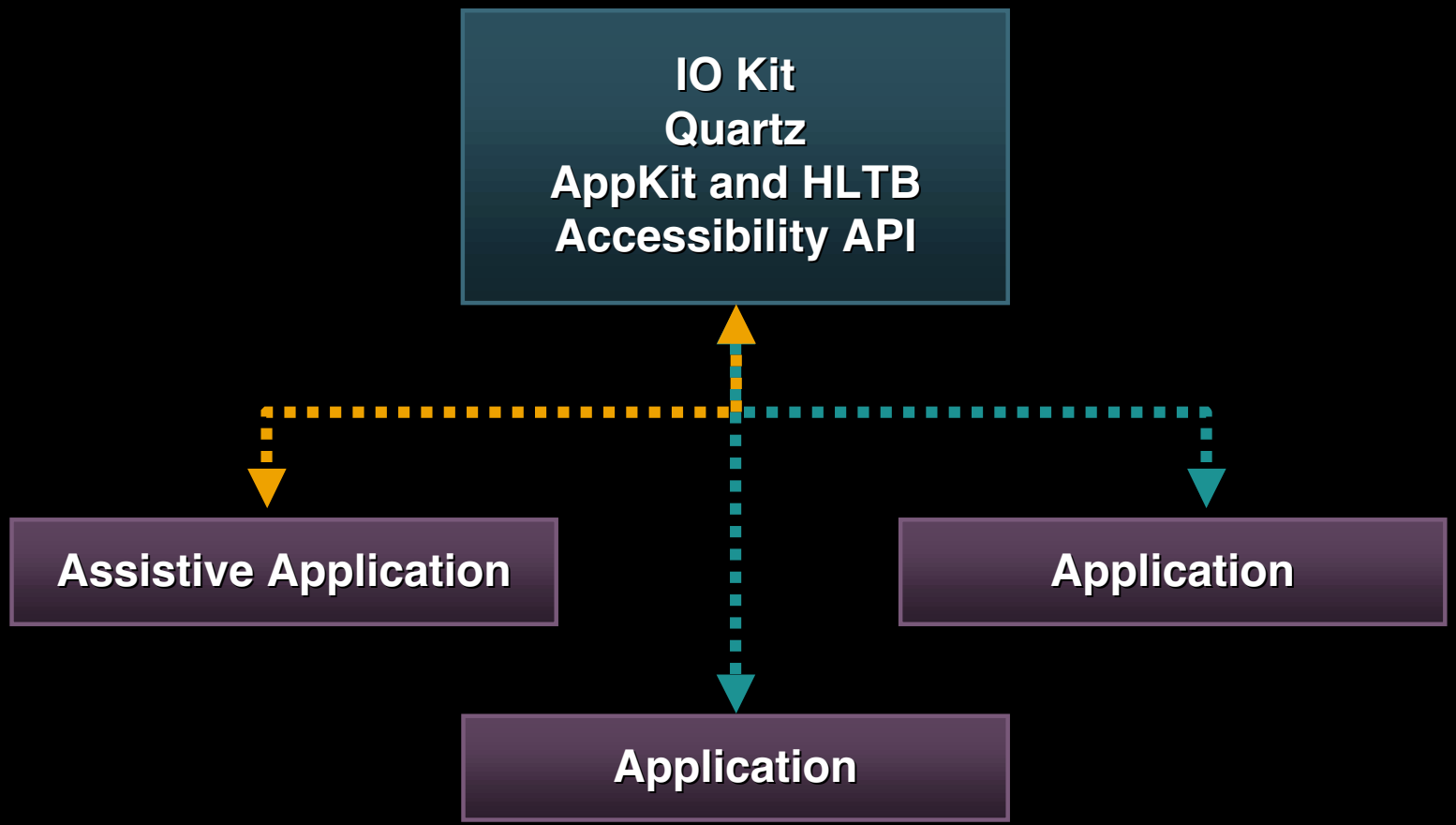
# Assistive Tech “Old Style”



# Assistive Tech Mac OS X Style




# Assistive Tech Mac OS X Style



# Managing Input

- High Level
  - Keyboard Navigation
- Mid Level
  - Quartz's **CGRemoteOperation** API
  - USB Human Interface Device (HID) Classes
- Low Level
  - IO Kit

# Discovering and Driving UI

- Jaguar Accessibility APIs
  - API-level support for Accessibility
  - Allows applications to discover and control UI in other processes
  - Provides common ground for assistive technology providers, application developers, and Apple

# Accessibility APIs

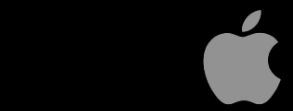
**Mike Engber**
**Cocoa Engineer**

# Accessibility—Overview

- Accessibility APIs
    - Examining other applications
- Clients
    - Assistive applications
- Targets
    - Cocoa and Carbon Applications
    - 209—Accessibility and Carbon
    - 304—Controls and Cocoa Accessibility

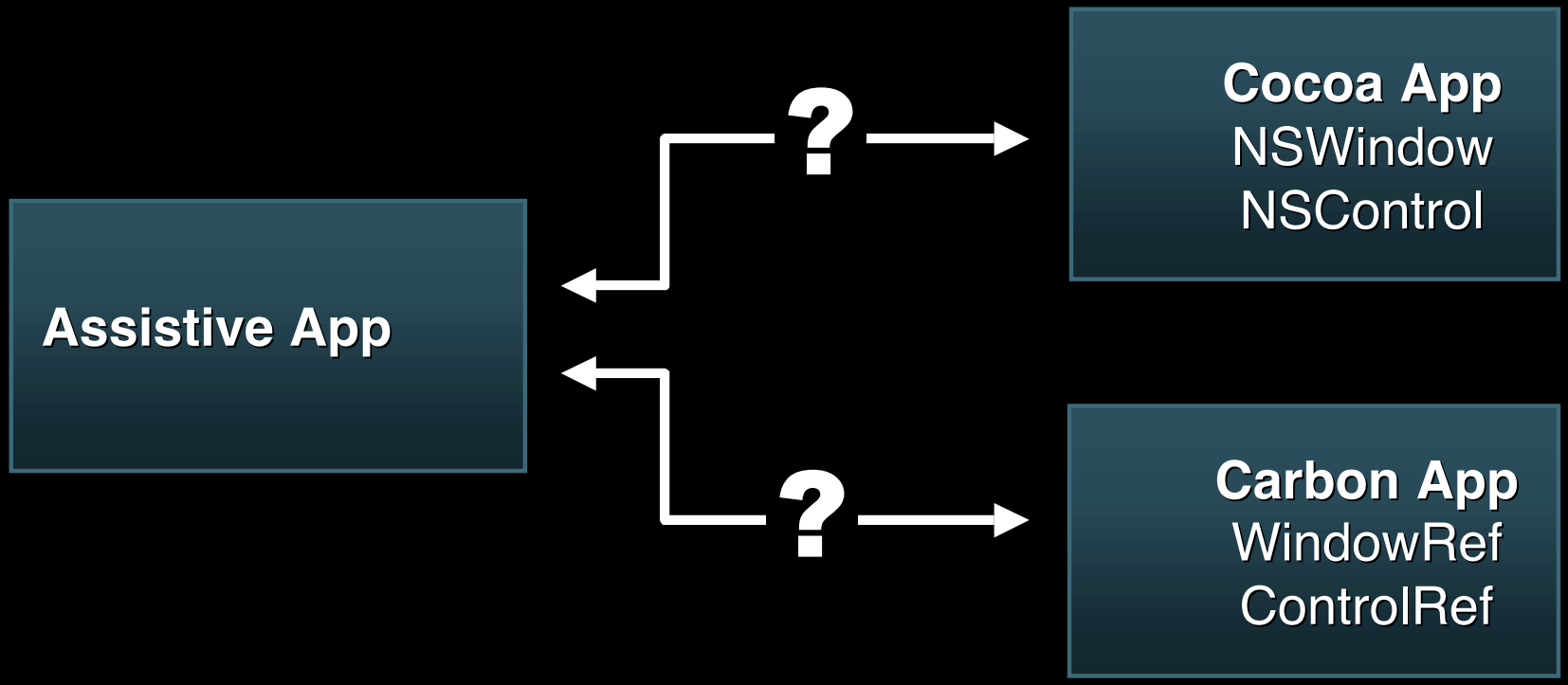
# Accessibility API Goals

- Assistive applications can:
  - Examine
  - Interact
  - Notifications
- Cocoa and Carbon targets
- Standard UI elements—just work
- Custom UI elements

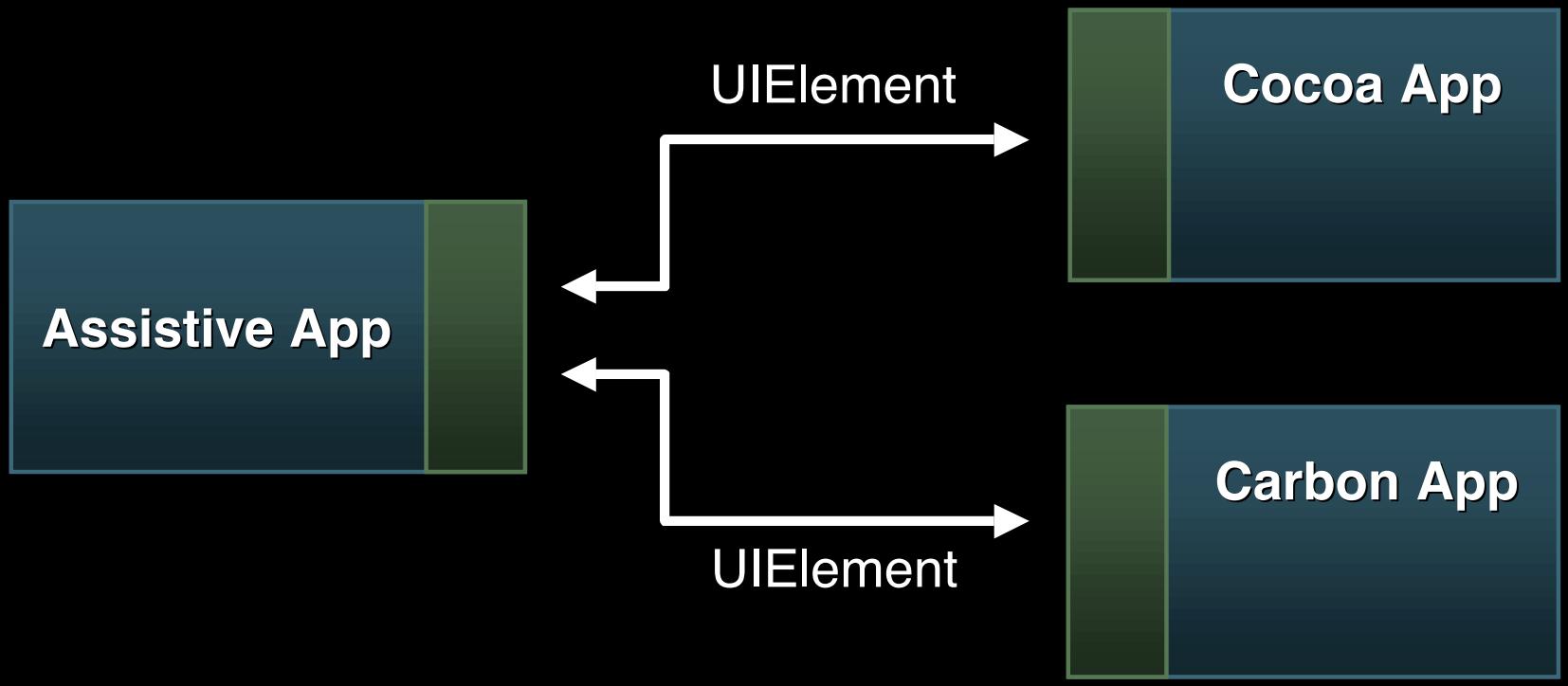


# Demo

# UI Representation—Problem



# UI Representation—Solution

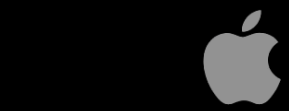


# UIElements

- Hierarchy of UIElements
- Attributes
  - **kAXTitleAttribute**
  - **kAXValueAttribute**
  - **kAXChildrenAttribute**
- Actions
  - **kAXPressAction**
  - **kAXIncrementAction**

# Attributes—Part 1

- Provide information
- Attribute Operations
    - Get a list of attributes
    - Get an attributes value
    - Test if an attribute can be set
    - Set an attribute's value



# Attributes—Part 2

- Role
  - Everyone has a role
  - Kind or class
- Role description
  - Localized string
- Title
  - Text in a button
  - “down” for an arrow

# Attributes—Part 3

- Value
  - Data type varies
  - Not always present
- Parent
  - Always present except at root
- Children
  - Traverse hierarchy
  - Not always present

# Actions—Part 1

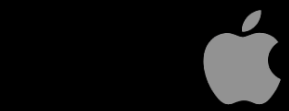
- Action ≈ mouse click
- Action Operations
    - Get a list of actions
    - Get an action's description
    - Perform an action

# Actions—Part 2

- Generic
  - “pick” or “press”
  - Not “Print” or “Cancel”
- Setting attributes often suffices
  - No select text action
  - Instead, set selected text range

# App/Top-level UIElement

- Starting point into the hierarchy
- Attributes
  - Children
    - Menubar
    - Windows
  - Focused UIElement
  - Active
  - Main/Key window



# Application UIElement—Part 2

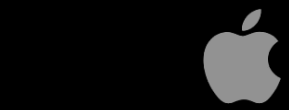
- Application element API
  - Hit testing
  - Posting keystrokes

# System-wide UIElement

- Represents all apps
- Supports
  - Hit testing
  - Posting keystrokes
  - Focused UIElement attribute

# Notifications—Part 1

- Window created, moved, miniaturized
- Application activated/deactivated
- UIElement destroyed
- Value changed



# Notifications—Part 2

- Specify a UIElement
    - Notifications about that element
- Specify the application
    - All notifications of that type

# What They Are Not

- Scripting
  - Drives the UI, not the app
- Programming environment
  - Only allow what a user could do
- Theme for MacHack 2002
  - Off by default

# Demo

# C APIs

Boolean **AXAPIEnabled** (void);

CFTypeID **AXUIElementGetTypeID** (void);

- Test if an object is a UIElement

```
If ( CFGetTypeID(ref) == AXUIElementGetTypeID() ) …
```

# C APIs—Top Level UIElements

```
AXUIElementRef  AXUIElementCreateApplication(pid_t pid);
AXUIElementRef  AXUIElementCreateSystemWide(void);

AXError AXUIElementCopyElementAtPosition(
                AXUIElementRef application,
                float x, float y,
                AXUIElementRef *element);

AXError AXUIElementPostKeyboardEvent(
                AXUIElementRef application,
                CGCharCode keyChar,
                CGKeyCode virtualKey,
                Boolean keyDown);
```

# C APIs—Attributes

```
AXError AXUIElementCopyAttributeNames(
                AXUIElementRef element,
                CFArrayRef *names);
AXError AXUIElementCopyAttributeValue(
                AXUIElementRef element,
                CFStringRef attribute,
                CFTypeRef *value);
AXError  AXUIElementIsAttributeSettable(
                AXUIElementRef element,
                CFStringRef attribute,
                Boolean *settable);
AXError  AXUIElementSetAttributeValue(
                AXUIElementRef element,
                CFStringRef attribute,
                CFTypeRef value);
```

# C APIs—Array Valued Attributes

```
AXError  AXUIElementGetAttributeValueCount(
                AXUIElementRef element,
                CFStringRef attribute,
                CFIndex *count);

AXError  AXUIElementCopyAttributeValues(
                AXUIElementRef element,
                CFStringRef attribute,
                CFIndex index,
                CFIndex maxValues,
                CFArrayRef *values);
```

# Data Types

C APIs—AXValues

- kAXValueCGPointType
- kAXValueCGSizeType
- kAXValueCFRangeType
- kAXValueIllegalType

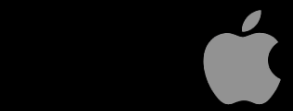
```
AXValueType AXValueGetType(CFTypeRef value);
Boolean AXValueGetValue(CFTypeRef value,
                        AXValueType theType,
                        void *valuePtr);
CFTypeRef AXValueCreate(AXValueType theType,
                        const void *valuePtr);
```

# C APIs—Actions

```
AXError  AXUIElementCopyActionNames(
                AXUIElementRef element,
                CFArrayRef *names);

AXError AXUIElementCopyActionDescription(
                AXUIElementRef element,
                CFStringRef action,
                CFStringRef *description);

AXError  AXUIElementPerformAction(
                AXUIElementRef element,
                CFStringRef action);
```



# C APIs—Notifications

```
AXError AXObserverCreate(
                pid_t application,
                AXObserverCallback callback,
                AXObserverRef *outObserver);

AXError AXObserverAddNotification(
                AXObserverRef observer,
                AXUIElementRef element,
                CFStringRef notification,
                void *refcon);

AXError AXObserverRemoveNotification(
                AXObserverRef observer,
                AXUIElementRef element,
                CFStringRef notification);


CFRunLoopSourceRef AXObserverGetRunLoopSource(
                                AXObserverRef observer);
```

# Documentation

## Accessibility

- Making Your Application Accessible to Users With Disabilities

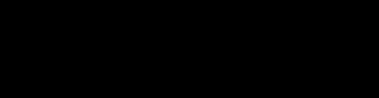
**ADC Member Site > Download Software > “Jaguar” Mac OS X > Docs**
connect.apple.com

- Accessibility Reference for Assistive Applications

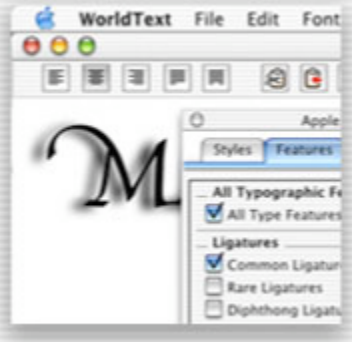
**“Jaguar” Mac OS X  Developer Tools CD**
/Developer/Documentation/ReleaseNotes/AssistiveAPI.html

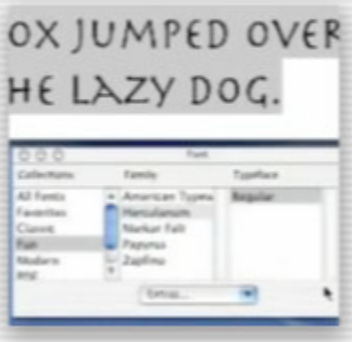
# Roadmap

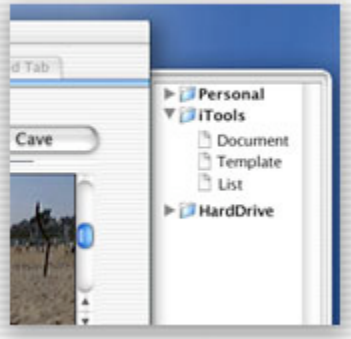
| | |
|---|---|
| **009 Accessibility Overview:**<br>Review the basic Accessibility concepts | Room A2<br>**Thurs., 2:00pm** |
| **209 Accessibility and Carbon** | Room A2<br>**Thurs., 3:30pm** |
| **304 Cocoa Controls and Cocoa Accessibility** | Room A2<br>**Thurs., 5:00pm** |



# Q&A


**Travis Brown**
**Graphics and Imaging Evangelist**
**Worldwide Developer Relations**

**http://developer.apple.com/wwdc2002/urls.html**


WWDC2002


WWDC2002


WWDC2002